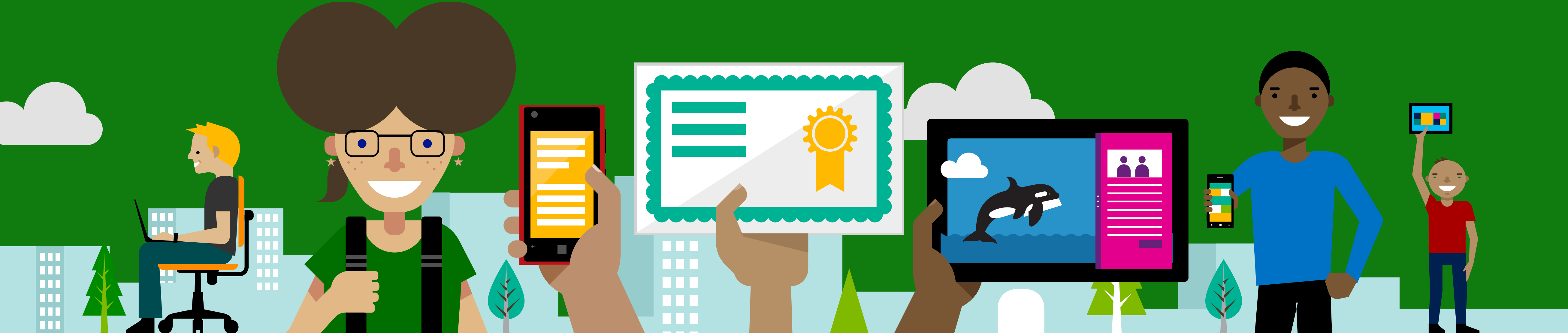
Microsoft

Managing Text Flow by Using CSS

3.3. Manage the flow of text content by using CSS.



Agenda

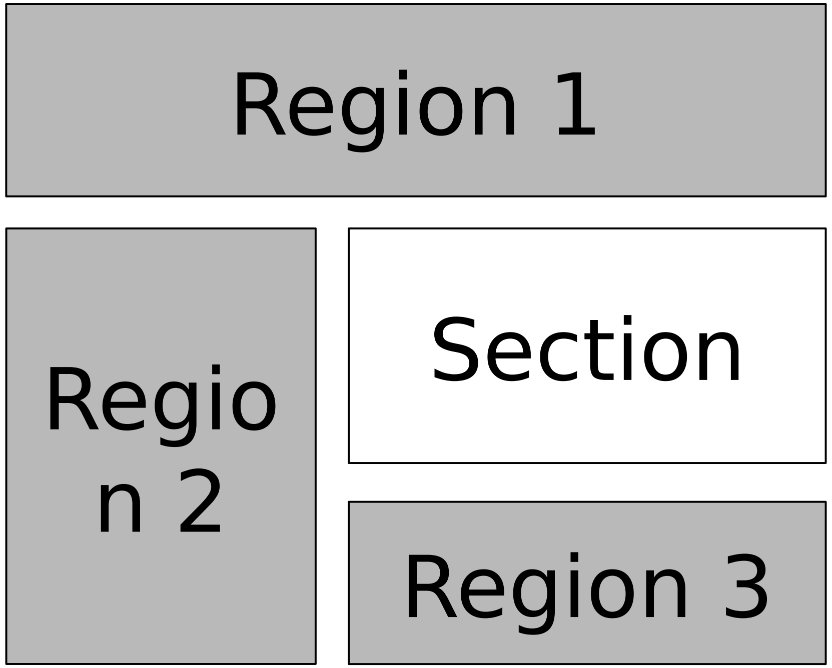
1 CSS Regions

2 Columns and Hyphenation

3 CSS Exclusions

CSS Regions

CSS Regions



- With HTML you can organize content into separate section, but what can you do if you want content to flow from one region to another?
- CSS Regions are sections of a document where content can flow from one region to another
  - If there is too much content in one section, then the remaining content automatically flows into another region
- Regions can be placed anywhere on a screen, as evidenced in the graphic to the right

Implementing CSS Regions

- Creating a CSS Region, you must specify a **content source** and **content containers**
  - You set up CSS regions with the `flow-into` and `flow-from` properties
- A content source is a chunk of text that holds the content you want to flow through the regions
- Content containers are the areas where you want your content to appear
  - You should use a separate HTML document as the source

```
<div id="content"
src="content.html"></div>
<div id="region1" class="region"></div>
<div id="region2" class="region"></div>
<div id="region3" class="region"></div>
```

`flow-from` and `flow-into` Properties

Using an id as a selector, identify the content source using the main value of the `flow-into` property

- This content will require content containers to appear on a Web page

With a specific class as the selector, use the same value used with the `flow-into` property for the value of the `flow-from` property to place content into a container

- This is how the source and content containers are associated

```
#content {
    -ms-flow-into: main;
}

.region {
    -ms-flow-from: main;
    background: #9ACD32;
    width: 150px;
    height: 200px;
    float: left;
    margin: 20px;
}
```

Overflowing Text

The height of a region won’t automatically adjust to fit all content

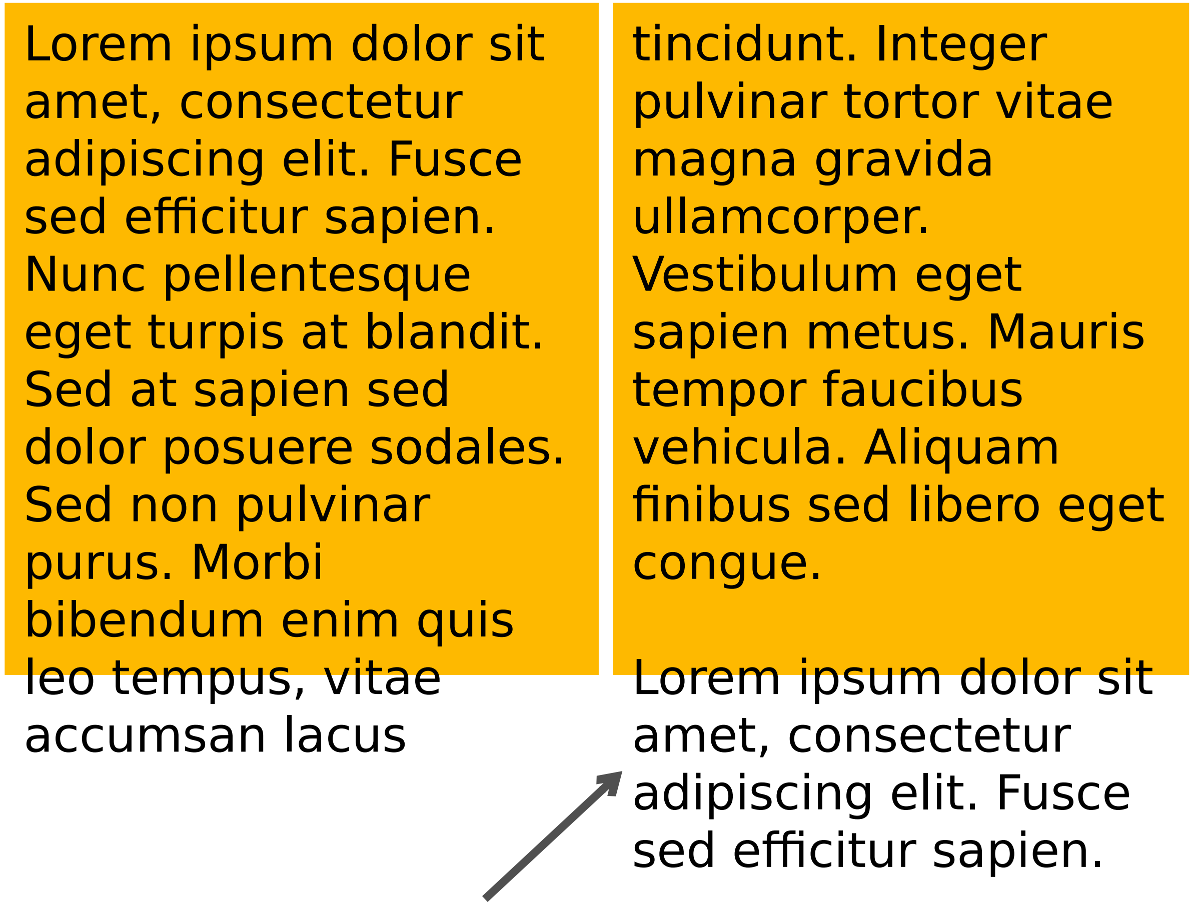
- height of regions must be defined

If there is content available after all regions have been filled, then the last region will:

- be cut off
- visibly overflow
- overflow and be hidden

The `overflow` and `region-fragment` properties can be used to control how overflow content is treated

- set `region-fragment` to `auto` or `break`



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sed efficitur sapien. Nunc pellentesque eget turpis at blandit. Sed at sapien sed dolor posuere sodales. Sed non pulvinar purus. Morbi bibendum enim quis leo tempus, vitae accumsan lacus

tincidunt. Integer pulvinar tortor vitae magna gravida ullamcorper. Vestibulum eget sapien metus. Mauris tempor faucibus vehicula. Aliquam finibus sed libero eget congue.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sed efficitur sapien.

visible overflow

The Microsoft Implementation of CSS Regions

- Instead of using `div` tags to implement CSS regions, Microsoft uses the `<iframe>` element
- An `iframe` is used with unique IDs, as well as the –ms– vendor prefix with the `flow-from` and `flow-into` properties

```
<iframe id="content"
src="content.html"></iframe>
<div id="region1" class="region"></div>
<div id="region2" class="region"></div>
<div id="region3" class="region"></div>
```

CSS Regions in Action

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <link href="StyleSheet.css" rel="stylesheet"
type="text/css">

        <title>Overflow Examples</title>
    </head>
    <body>
        <h1>CSS Regions</h1>
        <iframe id="content" src="content.html"></iframe>
        <div id="region1" class="region"></div>
        <div id="region2" class="region"></div>
        <div id="region3" class="region"></div>

    </body>
</html>
```

```
#content {
    -ms-flow-into: main;
}

.region {
    -ms-flow-from: main;
    -ms-region-fragment:
    auto;
    overflow: visible;
    background: #0ff;
    width: 150px;
    height: 200px;
    float:  left;
    margin:  20px;
}
```

Columns and Hyphenation

Creating Columns

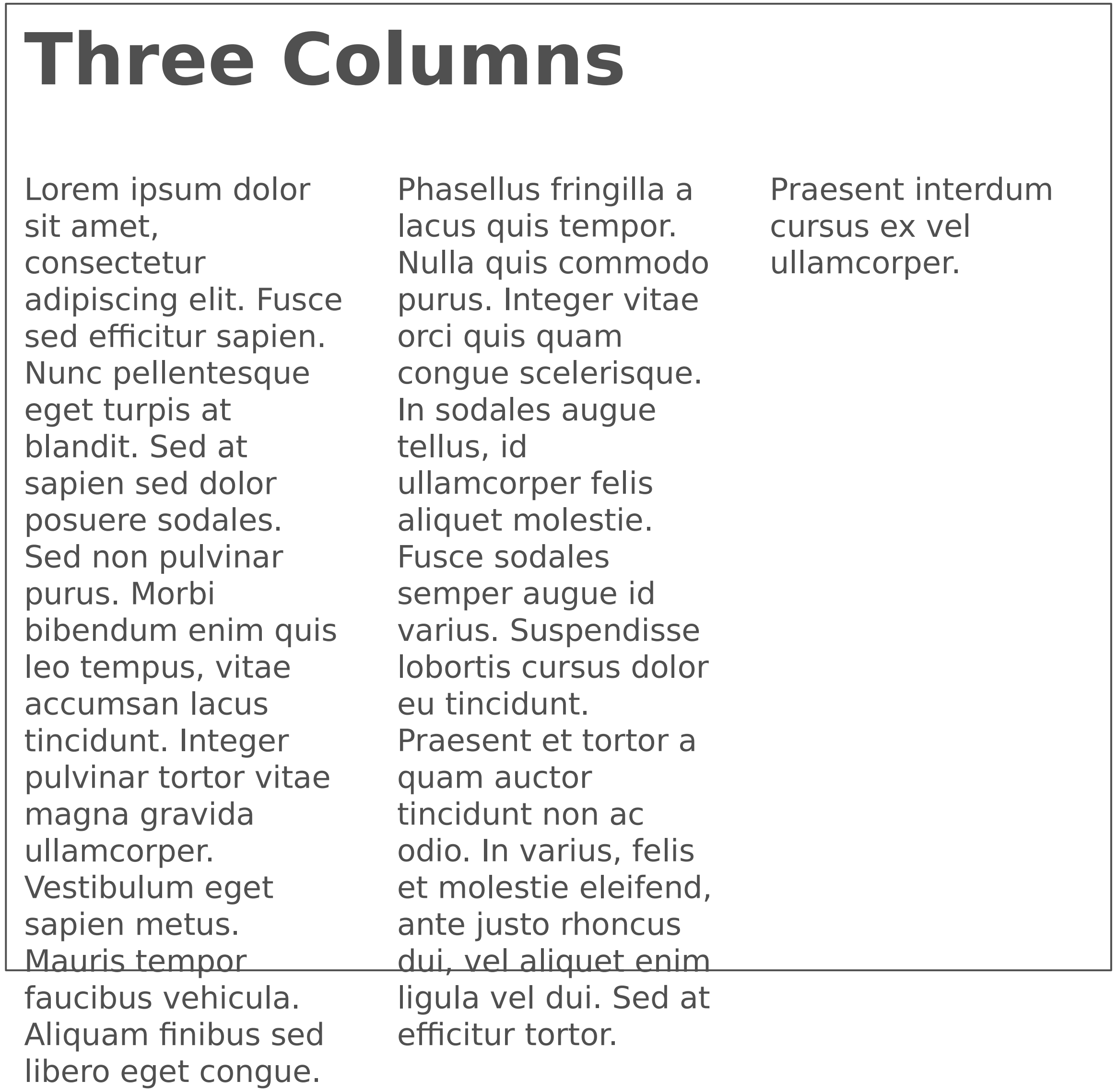
CSS allows you to create columns and divide content among them

- this is called the **multi-column layout**

There are three primary CSS properties that you can use to manipulate columns:

- `column-count`
- `column-gap`
- `column-rule`

In the CSS Box, columns are in between the content and the Box



Three Columns

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sed efficitur sapien. Nunc pellentesque eget turpis at blandit. Sed at sapien sed dolor posuere sodales. Sed non pulvinar purus. Morbi bibendum enim quis leo tempus, vitae accumsan lacus tincidunt. Integer pulvinar tortor vitae magna gravida ullamcorper. Vestibulum eget sapien metus. Mauris tempor faucibus vehicula. Aliquam finibus sed libero eget congue. Phasellus fringilla a lacus quis tempor. Nulla quis commodo purus. Integer vitae orci quis quam congue scelerisque. In sodales augue tellus, id ullamcorper felis aliquet molestie. Fusce sodales semper augue id varius. Suspendisse lobortis cursus dolor eu tincidunt. Praesent et tortor a quam auctor tincidunt non ac odio. In varius, felis et molestie eleifend, ante justo rhoncus dui, vel aliquet enim ligula vel dui. Sed at efficitur tortor. Praesent interdum cursus ex vel ullamcorper.

Column Properties, pt. 1

PROPERTY	VALUES	DESCRIPTION
break-after	auto always avoid left right page column region avoid-page avoid-column avoid-region	Inserts a break after the generated column box
break-before	(same as break-after)	Inserts a break before the generated column box

Column Properties, pt. 2

PROPERTY	VALUES	DESCRIPTION
break-inside	auto avoid avoid-page avoid-column avoid-region	Inserts a break within the generated column box
column-count	integer auto	Sets the number of columns an element will use
column-fill	auto balance	Specifies how to fill columns; balances content equally between columns, if possible, or fills columns sequentially
column-gap	length normal	Specifies the gap between columns

Column Properties, pt. 3

PROPERTY	VALUES	DESCRIPTION

`column-rule`	`column-rule-width`
	`column-rule-style`
	`column-rule-color`
	`transparent`
		Is a shorthand property that sets the column-rule-width, column-rule-style, and column-rule-color properties at the same place in a style sheet

`column-rule-color`	`color`	Specifies the color of the rule between columns
`column-rule-style`	`border-style`	Specifies the style of the rule between columns, such as solid or double, dashed, and so on
`column-rule-width`	`border width`	Specifies the width of the rule between columns
`column-span`	`none` `all`	Specifies whether an element should span no columns or all columns

Column Properties, pt. 4

PROPERTY	VALUES	DESCRIPTION
`column-width`	`length` `auto`	Specifies the width of a column or columns
`columns`	`column-width` `column-count`	Sets the column-width and column-count properties simultaneously

Using Hyphenation

- If you want text to cover the entirety of a column, then you can enable hyphenation
- Hyphenation is the process of connecting two parts of the same word with a hyphen (-)
- To implement hyphenation, use the hyphens property along with one of three values:
  - `auto`: enables hyphenation
  - `manual`: enables hyphenation
  - none: prevents hyphenation

```
.hyphenized {
    -moz-hyphens: auto;
    -ms-hyphens: auto;
    -webkit-hyphens:
auto;
    hyphens: auto;
}
```

Declaring a Language

- For automatic hyphenation to properly function, you have to declare which language is being used
- To do so, use the <html> tag with the `lang` attribute
  - The value should be set according to the language being used
- Vendor prefixes must also be applied with the hyphens property in CSS for them to work properly

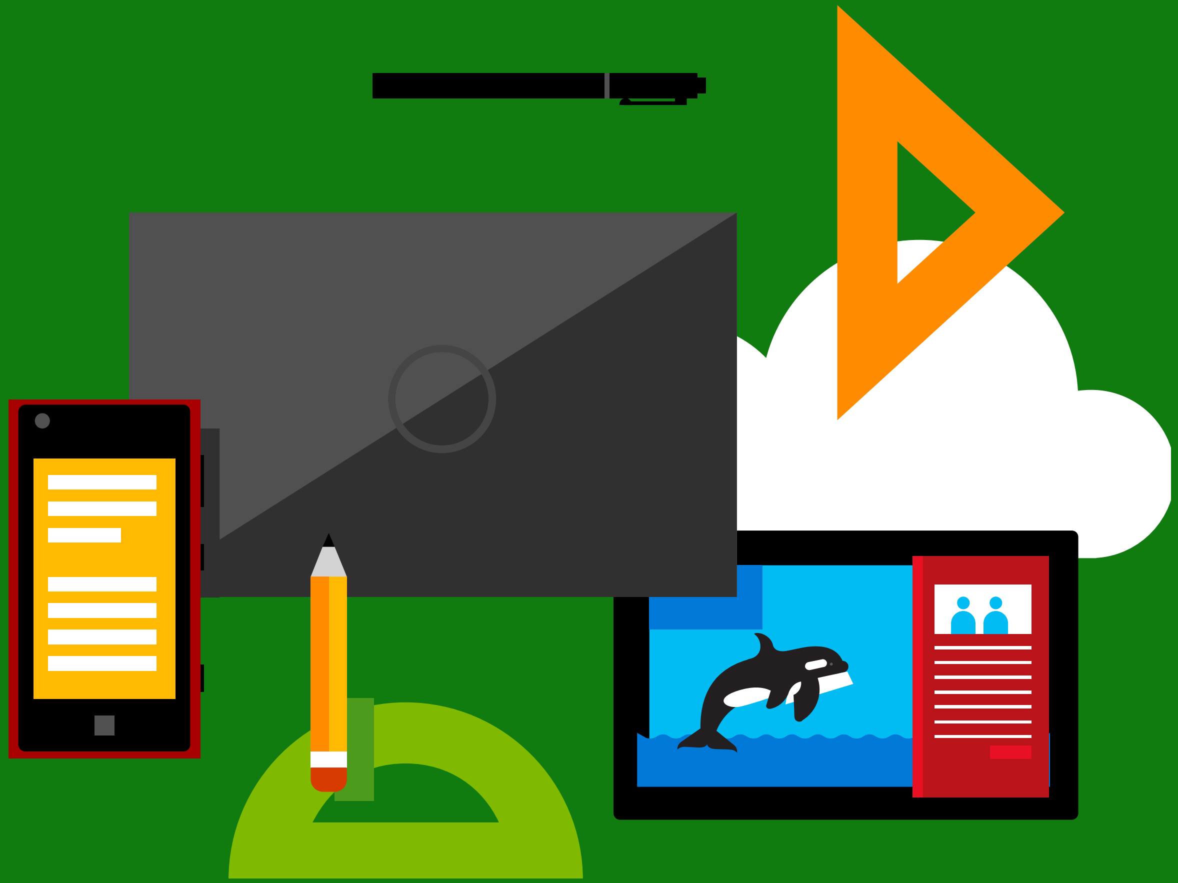
```
<html lang="en">
```

Columns Demo

```html
<h2>Three Columns</h2>
<div class="threeColumns”>
  <p>Lorem ipsum dolor …</p>
  <p>Phasellus fringilla a…</p>
</div>
```

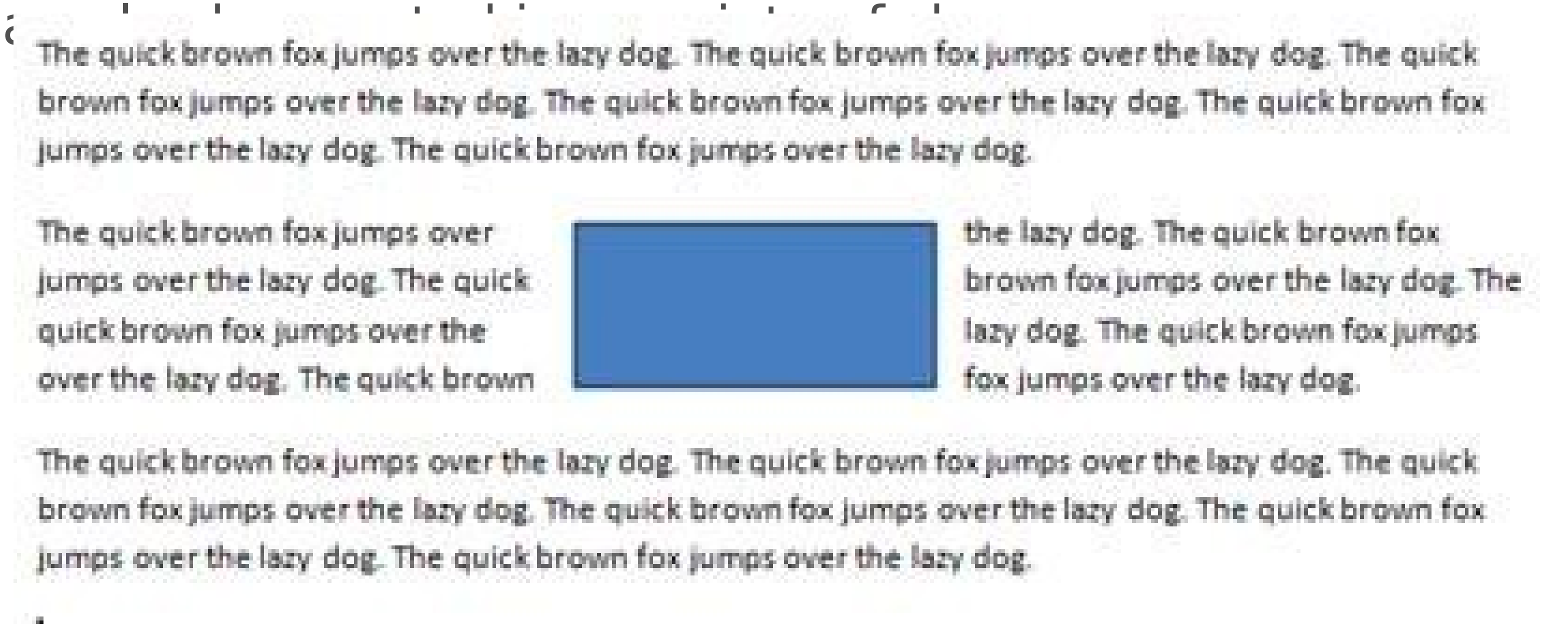
```css
.threeColumns {
    width: 100px;
    -webkit-column-count: 3;
    -moz-column-count: 3;
    column-count: 3;
}
```

CSS Exclusions



CSS Exclusions

- A **CSS Exclusion**, commonly referred to as a positioned float, allows you to place elements in a specific location on a Web page
- After you position the float, other content will wrap around it
- CSS Exclusions allow you to specify positioning from the top, bottom, left, and right of a container
  - Exclusions ca



The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over
jumps over the lazy dog. The quick
quick brown fox jumps over the
over the lazy dog. The quick brown

the lazy dog. The quick brown fox
brown fox jumps over the lazy dog. The
lazy dog. The quick brown fox jumps
fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

Creating CSS Exclusions

- Declare an exclusion by using the `wrap-flow` property with the preferred value
- Modify the shape of an exclusion using the `shape-outside` and `shape-inside` properties
- CSS Exclusions may not be supported by all of the major browsers
  - Use vendor prefixes and experiment with as many browsers as possible to make sure the content renders properly

```
.exclusion {
      -ms-grid-row: 2;
      -ms-grid-column: 2;
      background-color: lime;
      -ms-wrap-flow: both;
      padding: 10px;
      -ms-wrap-margin: 15px;
    }
```

Properties for CSS Exclusions, pt. 1

PROPERTY	VALUES	DESCRIPTION
shape-outside	auto shape url	Creates the general shape of an exclusion
shape-inside	outside-shape auto shape uri	Modifies a shape’s content
wrap	wrap-flow wrap-margin wrap-padding	A shorthand method of setting wrap-flow, wrap-margin, and wrap-padding properties in one declaration

Properties for CSS Exclusions, pt. 2

PROPERTY	VALUES	DESCRIPTION

`wrap-flow`	`auto`
	`both`
	`start`
	`end`
	`maximum`
	`clear`
		Specifies how exclusions affect inline content within block-level elements

`wrap-margin`	`length`	Provides an offset for content outside the element
`wrap-padding`	`length`	provides a pad (offset) for content inside an element
`wrap-through`	`wrap` `none`	Specifies how content should wrap around an exclusion element

More About CSS Exclusions

https://msdn.microsoft.com/en-us/library/hh673558(v=vs.85).aspx

Summary

1 CSS Regions

2 Columns and Hyphenation

3 CSS Exclusions

Microsoft